知識コミュニティにおけるエージェントの実装法

Implementation of Agents in the Knowledgeable Community

武田英明

奈良先端科学技術大学院大学 情報科学研究科

Graducate School of Information Science

Nara Institute of Science and Technology

The purpose of the Knowledgeable Community Project is to provide a knowledge-level cooperative framework for human and computers, e.g., to share, reuse, systemize, create knowledge. There are two major research topics in the Knowledgeable Community Project. One is knowledge media that provides appropriate ways to communicate human and computers according to contents of knowledge and environments. The other is hypermediator that integrate various kinds of information by required viewpoints.

Agent architecture is used to realize hypermediators in the Knowledge Community. We adopt KQML (Knowledge Query and Manipulation Language) as communication language and KIF (Knowledge Interchange Format) as knowledge exchange language, and Ontolingua as ontology description language. We built ontology-based mediators that handle inter-agent messages by interpreting messages based on ontology of messages and agents. We also provide inter-ontology translators based on multiple ontology mechanism. We built some application systems based on the above environments, i.e., KC-kansai that a pilot program for regional information system, Virtual tour that collects and arranges tour information, and a knowledge media system for technical documents.

1 はじめに

知識コミュニティ・プロジェクトは知識の共有・利用・体系化・創出のための、人間とコンピュータの知識レベルでの共同作業の枠組みを実現することを目的としている。 このために、二つの問題について研究を行なっている。一つは知識メディアという考えで、人間とコンピュータがその知識の内容や状況・環境に応じて適切な方法でコミュニケーションを実現するものである。二つめは一定の視点からいかに情報を集め、体系化して情報を提供するかという問題である。これを現在ハイパーメディエータと呼んでいる。ハイパーメディエータでは、情報を求めている人間・エージェントとその情報を持つ人間・エージェントを仲介する。この仲介においては、単に情報の所在の発見や形式の変換だけでなく、人間むきのメディアと機械向きのメディアの統合や、問題の解釈などが必要となる。現在、この問題をエージェント・アーキテクチャによって実現することを研究している。

本発表では、我々が用いているエージェント・アーキテクチャの実際とそれに基づくエージェントやシステムを紹介する。

2 エージェント・アーキテクチャの構造

我々はこれまで主に ARPA のエージェント・アーキテクチャに準拠したエージェントの構成を用いてきた (図1参照) 。

Facilitator 組織層としては facilitator を用いた連邦アーキテクチャ (federation architecture) を用いている。これはエージェント群を一定の単位でわけ、エージェント群各々に facilitator と呼ばれる特別なエージェントをおく。 facilitator の機能は

- メッセージ・マネージャ：自分の管理するエージェントが送受信するメッセージを中継する。メッセージ ID など付加情報をメッセージに加える。
- エージェント・マネージャ：エージェントの登録・起動・削除などを行なう。
- ネットワーク・マネージャ： 他の facilitator やネットワーク監視エージェントへ情報を提供する。

KQML 通信層はさらに 3 つの層からなっている。まず、メッセージの形式としては KQML(Knowledge Query and Manupulation Laguage)[1] を用いている。 KQML は発話行為をモデルとしたメッセージ行為のプロトコルである。 KQML ではメッセージは performative と呼ばれ、その種類 (performative type) によってそのメッセージを発した意味が指定される。また、同時に performative の中で許される項目の種類も規定される。 Performative type としては、 tell, ask-one, ask-about, ask-all, sorry などがある。また、項目としては、 sender (送り元)、 receiver(送り先)、 ontology (オントロジー)、 language (利用言語)、 content(内容) などがある。このうち、 language(利用言語) と ontology(オントロジー) はそれぞれ、content (内容) の構文と意味を規定する。

KIF 現在の知識コミュニティ KC0 では、利用言語として主として KIF[3][2] を用いている。 KIF(Knowledge Interchange Format) は知識の交換用の言語として提案されているもので、一階述語論理を基本とした記法によって各種の言語での表現を表現する言語である。

Ontolingua エージェント間で共有する概念体系の表現としては、 ontology として表現し、具体的には Ontolingua[4] を採用している。 Ontoligua は ontology を書くための言語で、 KIF の表現を拡張して、 class や関係を定義できるようにしている。 ontolingua 上でフレーム型のオントロジーを定義することで、クラス階層をもったオントロジーを記述することが出来る。

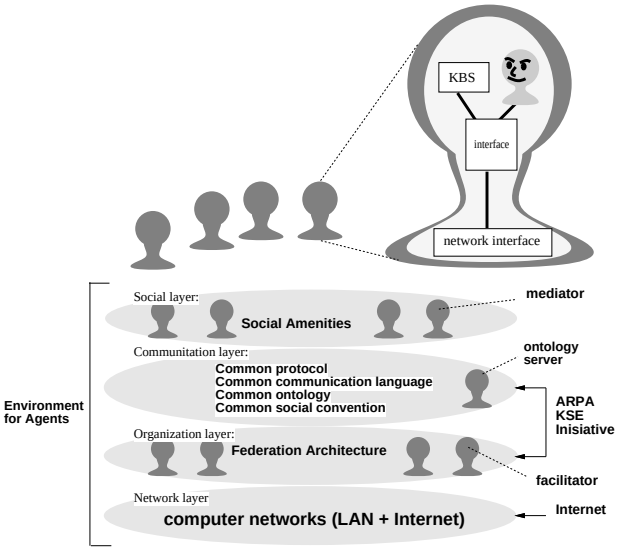

図 1: 知識コミュニティ KC0 におけるエージェント

3 エージェントの実現方法

ここでのエージェントは unix で KAPI(KQML Application Interface) を持つプロセスによって実現されている。 KAPI は先に述べた KQML の形式のメッセージを受け付けたり、逆にメッセージを渡すといった機能を提供するもので、「接続」「切断」「送信」「受信」など簡単なコマンド (関数) からなる。

KAPI で呼び出される通信方法はいろいろ可能であるが、ここでは mbus というプログラムを用いている。 MBus は unix のプロセス間でメッセージをやり取りするシステムである。マシン毎あるいはいくつかのマシン毎に"mbus"と呼ばれるデーモン・プロセスを走らせる。 MBus を通じて通信を行ないたいプロセスは、まず mbus デーモンに対して自分を登録する。この後はこのプロセスは自分が mbus を通じて他のプロセスに対してメッセージを送ることができ、また他のプロセスからメッセージを受けとることができる。接続を切りたい時は、プロセスは、登録を抹消することを mbus デーモンに伝える。現在、 mbus 用 KAPI は Common Lisp 用、 emacs lisp 用、 tcl/tk 用、 prolog 用を開発している。

KAPI を持つプロセスがエージェントとなりうるが、1 プロセスが 1 エージェントであるとは限らない。我々は Common Lisp 上でエージェント・プロトタイプ用言語を用意したが、そこでは 1 つのプロセスで複数のエージェントを動かすことができる。 KAPI から見た時 1 つにみえる。 KAPI/facilitator に登録された名前が同一であるものが一つのエージェントである。

4 Ontology に基づくエージェント組織

複数のエージェント間で知識交換を行なうためには、その知識の背景にあるオントロジーをエージェント間で共有していなければならない。知識コミュニティではオントロジーを管理するエージェント (ontology server) を用意している。 Ontology server はエージェント間で共有される概念体系の記述の他、エージェントとエージェントの持つ知識の背景となる概念との関係を保持している。この知識を元にメッセージが与えられた時メッセージを受理可能なエージェントを推定することができる。これをオントロジーに基づくメディエーションと呼んでいる [5]。さらにオントロジーが多重化したときは、オントロジーの記述をたどっていくことによりメッセージの変換を行なうことができる [7]。

5 実験システム

これまで述べてきたような枠組みに基づいていくつかのシステムを試作してきた。

- KC-Kansai：地域情報システムのプロトタイプ。
- Virtual Tour：断片的な知識を用いて、旅行のイメージを構成する。
- 専門文書知識メディア・システム：専門書をオントロジーを元に再構成して提示する。
- GuardNet：エージェントで構成されたエージェント開発環境 [6]。

6 議論

これまでの研究のなかで特に問題となった点は以下の点である。

- オントロジー構築の難しさ：オントロジーはその目的や意図によって求めるものが異なり、そのたびにオントロジーを構成するのは困難である。
- 人間の理解とのギャップ：オントロジーやメッセージを複雑化すると人間の理解と乖離する。

前者に対しては、ボトムアップな情報構造化や学習といったことが必要であると思われる。後者に対しては知識メディアの取り込みがひとつのアプローチであろう。

参考文献

[1] Tim Finin, Don McKay, Rich Fritzson, and Robin McEntire. KQML: An information and knowledge exchange protocol. In Kazuhiro Fuchi and Toshio Yokoi, editors, *Knowledge Building and Knowledge Sharing*. Ohmsha and IOS Press, 1994.

[2] M.R. Genesereth. Knowledge interchange format. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pp. 599–600. Morgan Kaufmann, 1991.

[3] M.R. Genesereth and R. E. Fikes. Knowledge interchange format, version 3.0 reference manual. Technical Report Technical Report Logic-92-1, Computer Science Department, Stanford University, June 1992.

[4] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.

[5] 武田英明, 飯野健二, 西田豊明. 知識コミュニティにおける仲介機能. マルチエージェントと協調計算 III, pp. 49–58. 近代科学社, 1994.

[6] 鷹合基行, 武田英明, 西田豊明. マルチーエジェント系の共同開発のためのマルチエージェン ト系. 人工知能学会全国大会 (第 9 回) 論文集, pp. 323–326, 1995.

[7] 武田英明, 飯野健二, 西田豊明. 構成的オントロジーによるオントロジーの多重化と分散化. 人工知能学会全国大会 (第 9 回) 論文集, pp. 367–370, 1995.